JPA-Repository als EJB

Das Repository auf Basis des HR-Schemas für Employees soll per Stateless (oderSingleton) EJB realisiert werden. Es kann die eigene oder als Basis die bereit gestellte Variante verwendet werden.

- Im Wildfly ist die externe H2-Database zu verwenden.
- Der EntityManager sollte per Dependency Injection beschafft werden. Welche Klasse realisiert den EntityManager?
- Die Transaktionsteuerung kann durch den EJB-Container erfolgen. Verwenden Sie deklarative Steuerung.
- Testen Sie die Funktionalität via Remote Client
- Optional: Ergänzung um Lifecycle-Methoden, Bean-Validation im JPA-Layer, ...

Die Beispiel-Kontoverwaltung soll als Singleton-EJB realisiert werden. Ist das Multi-Threading-Thema zufriedenstellen gelöst?

- Optional: Simulation der Last durch Client-seitigen Thread-Pool.
- Optional: Interceptor, der die Aufrufe protokolliert und den laufenden Thread ausgibt.

CDI: Kann anhand des MusikerBeispiels via CDI (Weld) auch ein Orchester bestehend aus Gitarristen und Klavierspielern erstellt werden. Es reicht dies in der SE zu prüfen.

CDI: Erzeugen Sie in ener Servlet-WebAnwendung jeweils eine CDI.Bohne für die 3 Standardscopes (Application, Session, Request). Selbsverständlich kann sich die kurzlebigen auch die langlebigen spritzen lassen. Was passiert wenn die Application-Bean sich diue Session-Bean spritzen lässt und die Session-Bean die Request-Bean?

 Testen Sie dies, indem das Servlet sich die Application-Bean spritzen lässt und via HTTP-Get entsprechende Ausgaben erzeugt. Z.B. die passende toString-Methoden der Beans aufgerufen werden. Optional CDI: Die Kontoverwaltung kann ebenso wie das JPA-Repository via CDI realisiert werden. Allerdings benötigt man für den Zugriff von aussen ein Web-Frontend oder eine EJB. Z.B. könnte eine EJB mit Remote-Interface als Fassade dienen.